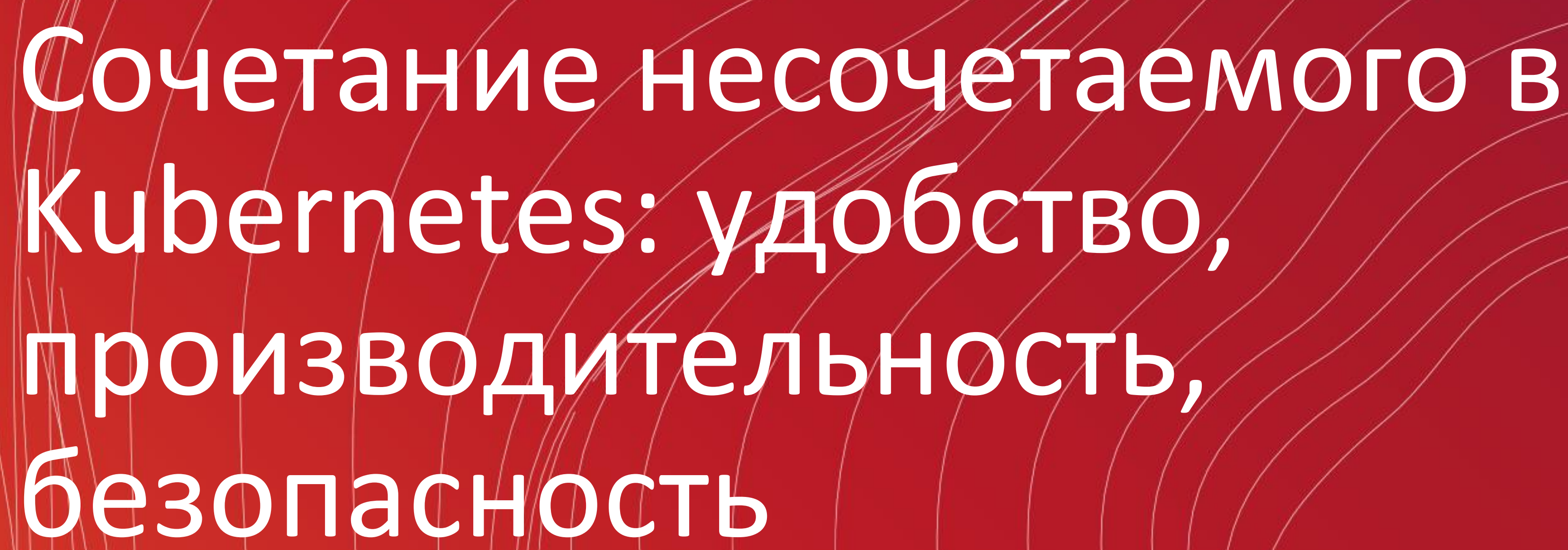


Сочетание несочетаемого в Kubernetes: удобство, производительность, безопасность

Евдокимов Дмитрий

Founder&CTO Luntry





Обо мне

- Основатель и технический директор Luntry
- Опыт в ИБ более 10 лет
- CFP ZeroNights, DevOpsConf
- Автор статей и бывший редактор рубрик в журнале “ХАКЕР”
- Автор Telegram-канала “k8s (in)security”
- Автор курса “Cloud Native безопасность в Kubernetes”
- Не верю, что систему можно сделать надежной и безопасной, не понимая ее
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, OFFZONE, PHDays, Kazhackstan, DevOpsConf, KuberConf, VK Kubernetes Conference, HighLoad++ и др.



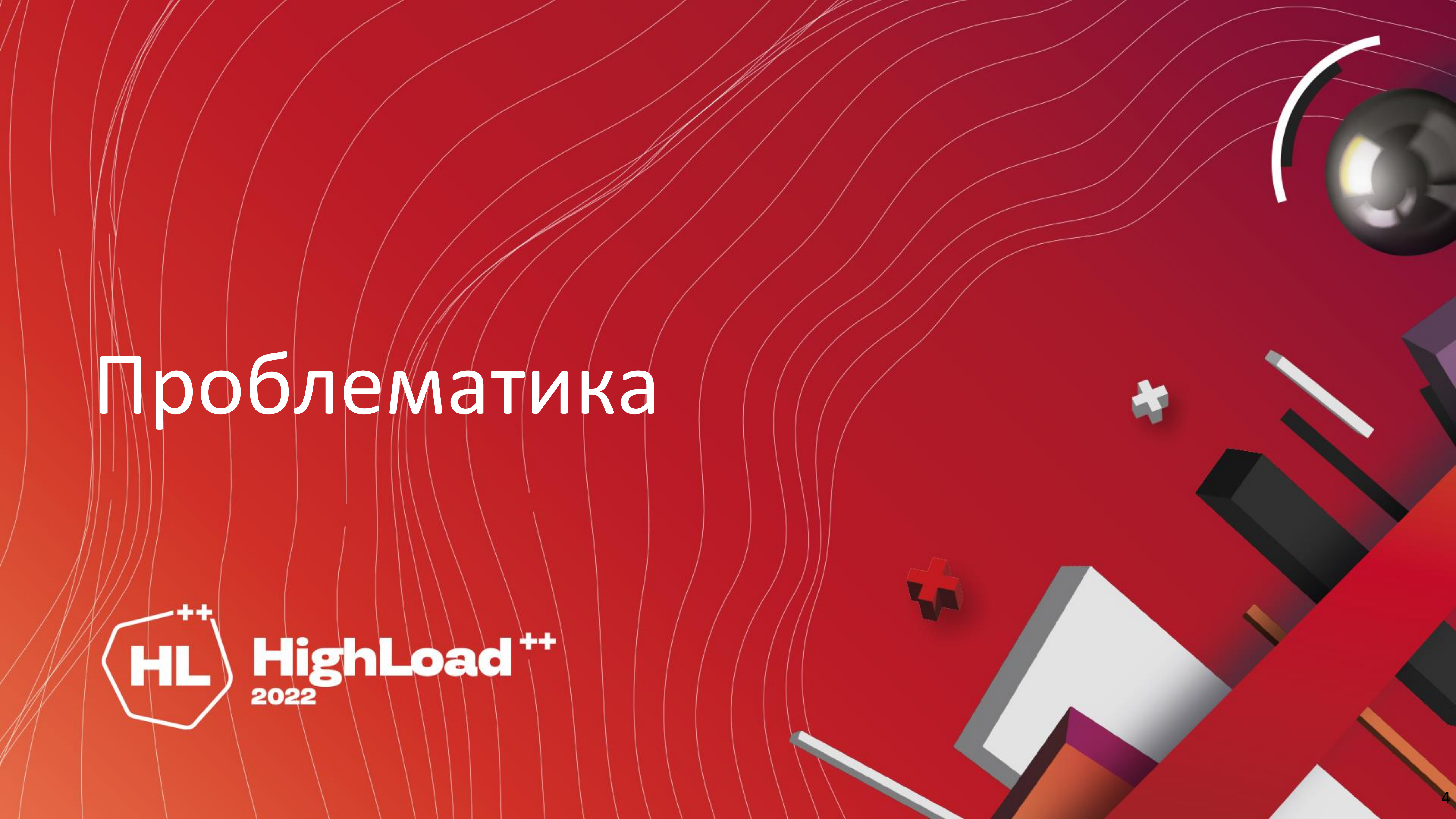




Содержание

- Проблематика: ИТ vs ИБ
- Проблемы и решения
  - На уровне кластера
  - На уровне ОС хоста
  - На уровне базовых образов контейнера
  - На уровне сети
- Заключение




Проблематика
HL
HighLoad++
2022



Security vs Usability

ИБ



ИТ





Чем мы будем сегодня заниматься ?



в Kubernetes ;)





Что хочет ИТ и ИБ?

- ИБ: Отсутствия нарушений конфиденциальности, целостности, доступности информации. Контроля рисков, угроз, уязвимостей. Повышения цены атаки.
  - Как привыкли делать: создавать/брать уже готовые чек-листы, формировать требования и заводить в рамки ИТ
- ИТ: Наличия быстрой, эффективной, экономичной, надежной системы.
  - Как привыкли делать: искать обходные пути требованиям ИБ*, страдать или забивать

* - если она вообще есть в компании ;)





Примеры проблем

- Разбор гигантского количества уязвимостей с большим количеством false-positive-срабатываний
- Внеплановое/экстренное исправление уязвимостей с дополнительными временными и человеческими трудозатратами
- Сложные процедуры получения доступа и выкатывания приложений
- Ухудшение производительности и потребления ресурсов за счет дополнительных средств безопасности и иных средств контроля
- …





Общие интересы

- Не тратить время друг друга
- Не грузить друг друга муторной работой
- С пользой брать вычислительные ресурсы
- Выровнять скорость обеспечения безопасности до скорости доставки нового value до клиента




Кластер
HL
HighLoad++
2022



Проблемы на уровне кластера

- Кластер на команду
  - Высокий уровень изоляции и обособленности
  - Высокая цена
- Зоопарк кластеров
  - У каждой команды свой уникальный кластер
  - Сложно контролировать и поддерживать
  - Configuration drift



Платформа на Kubernetes



- Компании начали делать платформы, дистрибутивы на базе Kubernetes
  - Kubernetes – это ядро Linux XXI-го века
- Multitenancy для изоляции и сегментации
  - Cluster-as-a-Services
  - Namespace-as-a-Services
  - ControlPlane-as-a-Services
  - Node-based isolation
- Примеры проектов: Cluster API, KubeSlice, Capsule, Kamaji, vCluster, Kiosk, kcp, KubePlus

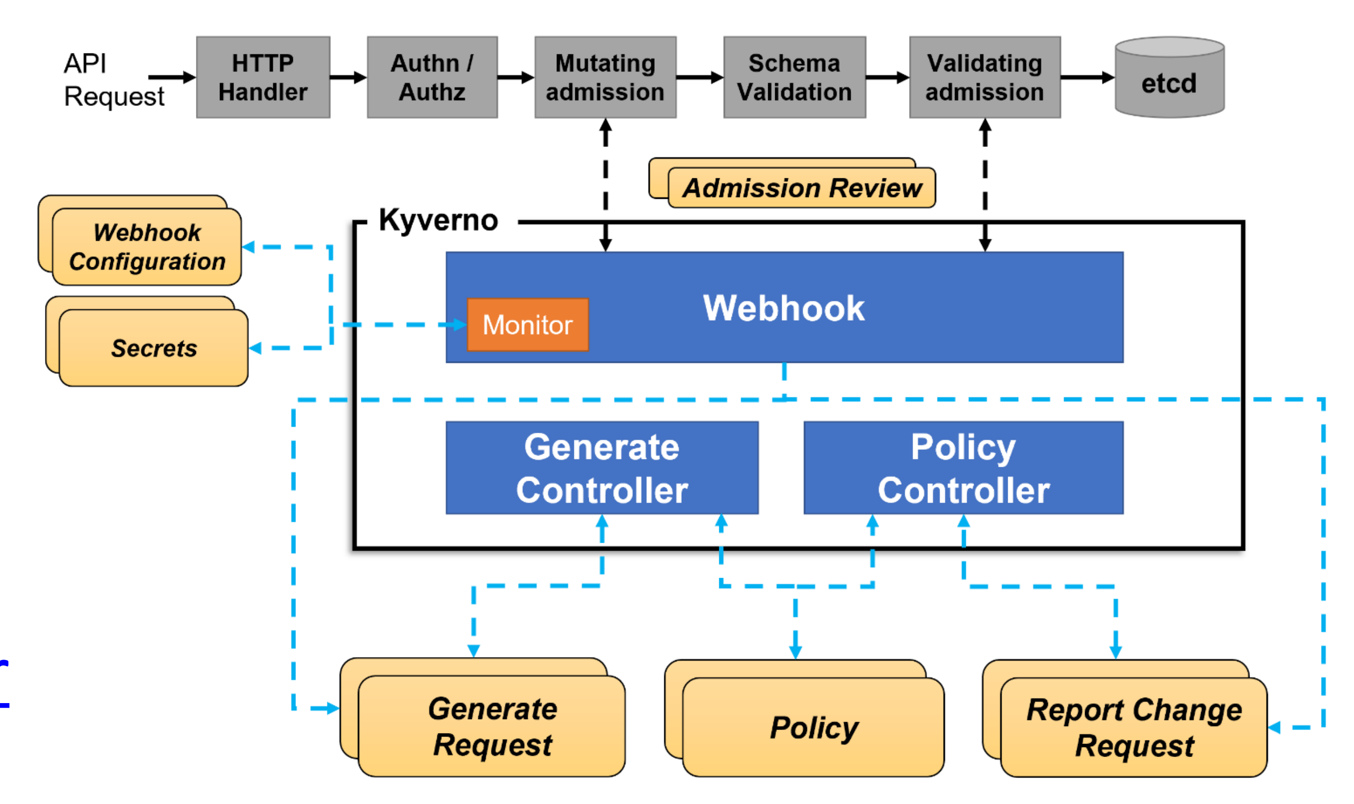




Day-2 поддержка

- Policy Engines
- Policy-as-Code
- Предупреждение и предотвращение
- Мутация и валидация ресурсов

- Пример: Kyverno, Gatekeeper





ОС на хосте



Проблемы на уровне ОС хоста



- Использование ОС общего назначения в Kubernetes кластерах: Ubuntu, Debian, CentOS, Fedora, …

Получаем:

- Разный цикл обновлений
- Большая поверхность атаки
- Много возможностей для атакующего
- Много шума от сканеров уязвимостей
- Много compliance требований и контролей (доступ, целостность, …)
- Configuration drift





Специализированные ОС для контейнеров

- Созданы для оркестраторов, а не для человека
  - Отсутствие привычных доступов
    - Отсутствие shell – исключает ряд рисков и угроз
    - Уменьшение вероятности configuration drift
- Поддержка оптимальной работы с контейнерами из коробки
- Минимальный размер
  - Высокая скорость разворачивания
  - Убрано все лишнее
    - Меньше false-positive-срабатываний сканеров
- Повышенные требования по ИБ
  - Частые обновления
  - Специальные патчи и конфигурации для ядра
  - ReadOnly-файловая система
  - …









Talos



Fedora CoreOS



Bottlerocket



Flatcar Container Linux



Container-Optimized OS (COS)



OS Talos

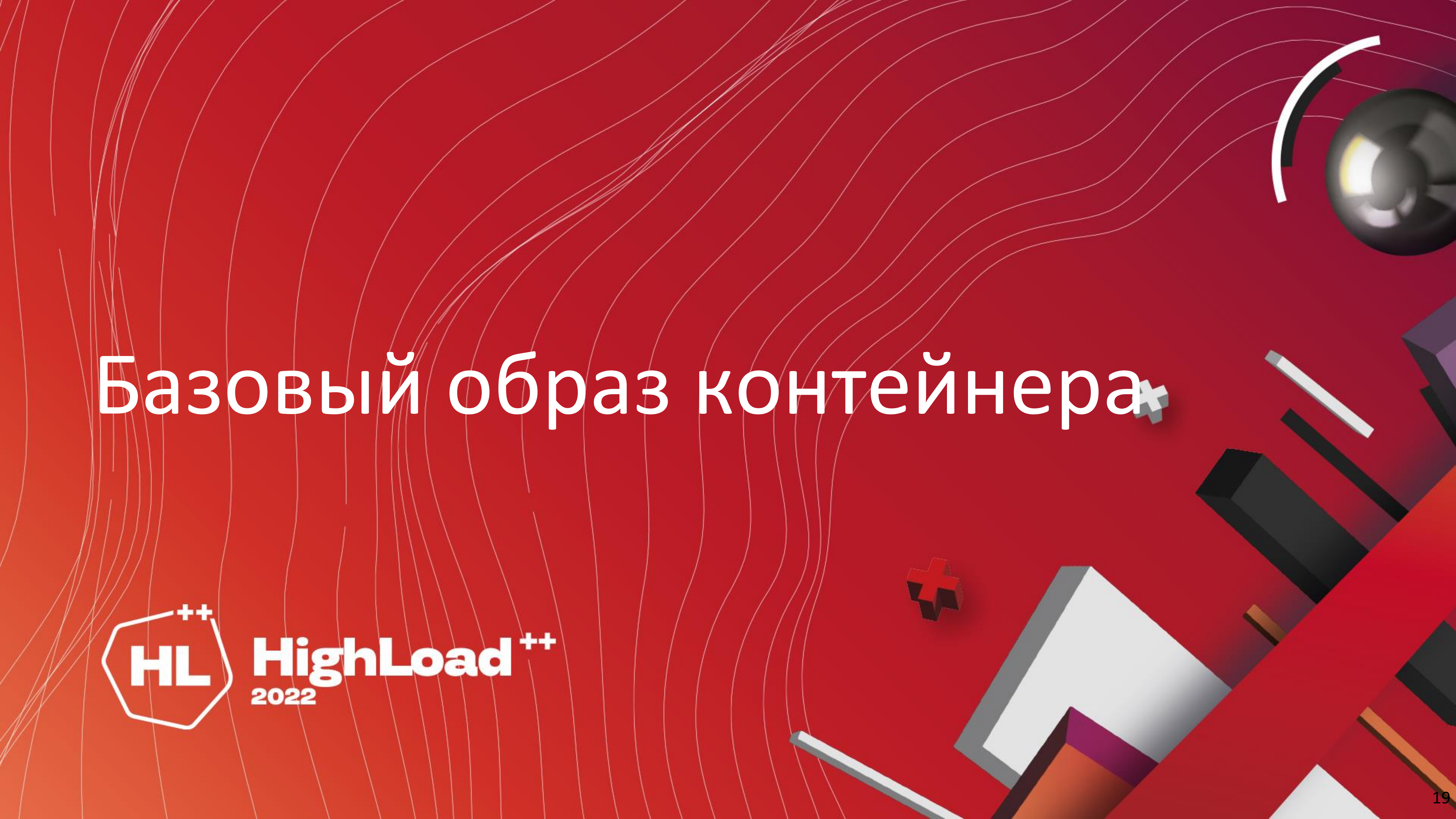


What is Talos Linux?

Talos Linux is Linux designed for Kubernetes – secure, immutable, and minimal.

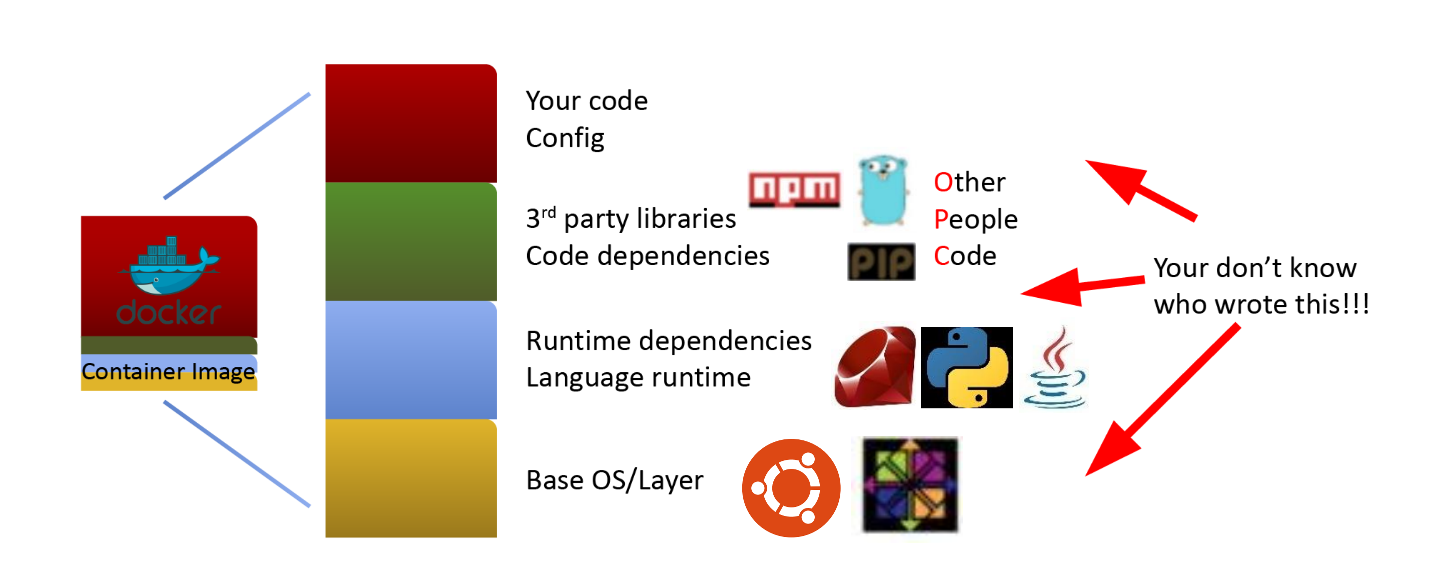
- Supports cloud platforms, bare metal, and virtualization platforms
- All system management is done via an API. No SSH, shell or console
- Production ready: supports some of the largest Kubernetes clusters in the world
- Open source project from the team at Sidero Labs

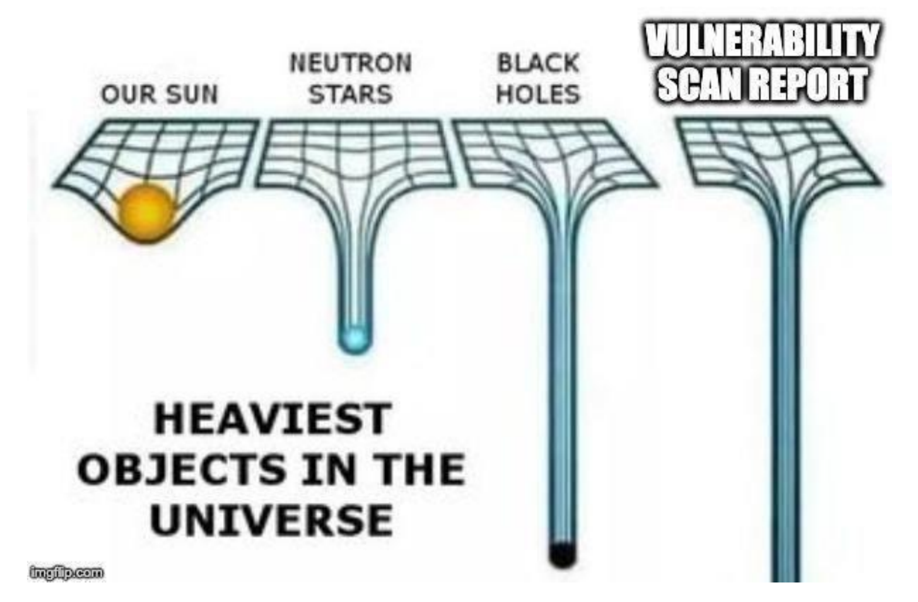



Базовый образ контейнера
HL
HighLoad++
2022

Проблемы на уровне базовых образов контейнеров







При этом помним, что Vulnerabilities != vulnerable, и это сильно мешает …





Проблемы сканеров уязвимостей

- ОС содержат много лишних, неиспользуемых файлов, которые расширяют поверхность атаки (например, shell)
- Даже в самых последних версиях образов ОС есть много известных уязвимостей
- Все сканеры выдают разные результаты, и их качество зависит от объекта сканирования
  - Отдельная большая болезненная тема, что и как делать с этими результатами
  - "Testing Docker CVE Scanners" [part 1, part 2, part 2.5, part 3]



Tiny/slim/minimal images



- Debian Slim – тонкий образ ОС Debian (debian:<suite>-slim).
- Alpine – маленький образ на базе ОС Alpine Linux с полным индексом пакетов
- Scratch – пустой образ для построения base images или для образов из одного “static” бинаря
- Distroless – образы от Google, ориентированные на определенный язык программирования, без ОС
- Chainguard Images – набор инструментов (apko, melange, wolfi) и образов для создания минималистичных distroless-образов

- DockerSlim – минификатор образа на основе поведения приложений в образе





Не все distroless одинаковые

"[What's Inside Of a Distroless Container Image: Taking a Deeper Look]"



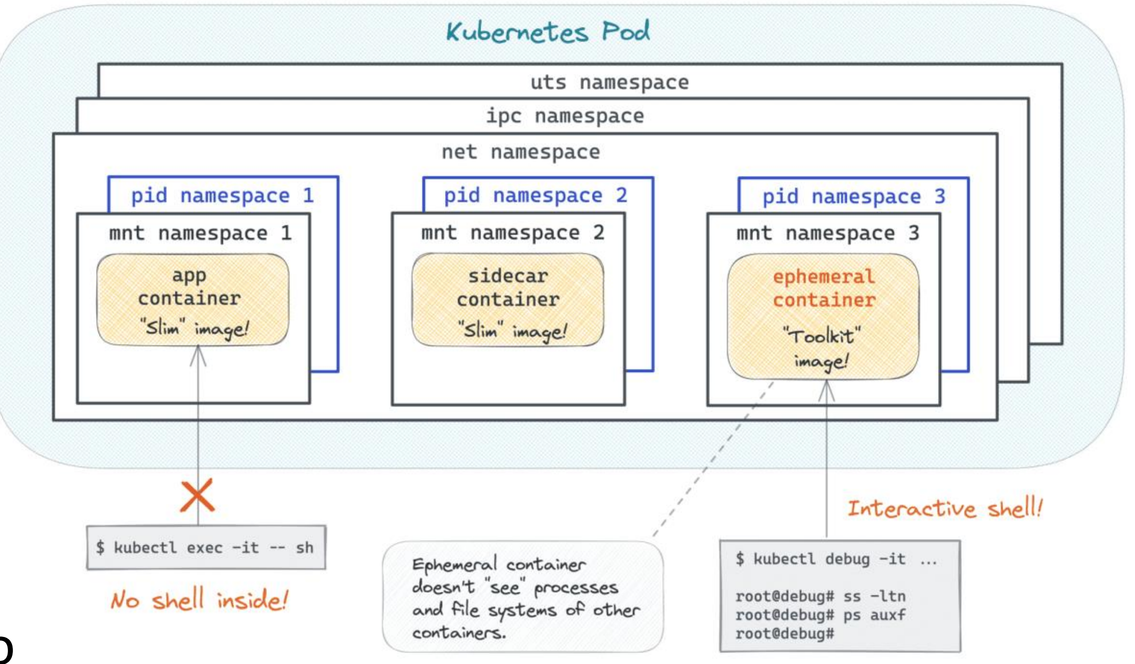


Отладка с помощью Ephemeral Containers

История:
- В 1.18 в Alpha
- В 1.23 в Beta
  - Включены по умолчанию
- В 1.25 в Stable

Режимы работы:
- kubectl debug в сочетании с shareProcessNamespace
- kubectl debug с указанием конкретного container в Pod
- kubectl debug с копированием целевого Pod

Пример образа Koolkits:
- Node.js
- Python
- Go
- Java



"Kubernetes Ephemeral Containers and kubectl debug Command"





Атаки на distroless-образы

- Эксплуатация в контейнерах на базе gcr.io/distroless/base
  - Позволяет читать, писать произвольные файлы и выполнять команды
  - Благодаря interactive command prompt от OpenSSL

```
% docker exec -it demo /bin/sh
OCI runtime exec failed: exec failed: container_linux.go:380: starting
container process caused: exec: "/bin/sh": stat /bin/sh: no such file or
directory: unknown
```

```
% docker exec -it demo /usr/bin/openssl
OpenSSL>
```

"Exploiting Distroless Images"





Chainguard Images

alpine Linux

Wolfi

melange

Custom apks

Custom apks

Alpine apks

apko

Wolfi apks

Alpine-based images

Wolfi-based images

build base
input
output

"apko Overview"

HighLoad++ 2022

Сравнение



Базовый образ	Размер	CVE*	Комментарий
Debian	~118Mb	74(2/10/1/61)	
Ubuntu	~69Mb	21(0/2/6/13)	
Debian Slim	~74Mb	74(2/10/1/61)	Убрали часть файлов типа «man pages» и «documentation»
Alpine	~7Mb	0(0/0/0/0)	Построен на musl libc и BusyBox
Scratch	0Mb	0(0/0/0/0)	
distroless/static	~2Mb	0(0/0/0/0)	Трудно модифицировать на Bazel
distroless/base	~20Mb	13(0/0/2/11)	Трудно модифицировать на Bazel
distroless/cc	~23Mb	13(0/0/2/11)	Трудно модифицировать на Bazel
distroless/cc + runtime	~54Mb	56(2/9/10/35)**	Поддержка nodejs, java, python3
chainguard	~94Mb	0(0/0/0/0) **	Легко модифицировать, часто обновляется, есть SBOM, есть подписи

** Данные получены сканером Trivy на 18.11.2022*
*** Зависит от runtime/содержимого (результат для python)*




Сеть
HL
HighLoad++
2022

Проблемы на уровне сети



- Отсутствие observability происходящего
  - Непонимание, что и как ограничивать
- Страх негативного влияния на производительность
  - NetworkPolicy устроит просадки
- Проблема не в получении доступа, а в отсутствии контроля над ней
  - Пока еще мало кто использует NetworkPolicy
- Сложно понять, кто и куда может обращаться, а куда нет
  - Когда NetworkPolicy становится много



Сетевая безопасность



Порядок применения NetworkPolicy на Cilium:

1. L3+L4 policy
2. L4-only policy
3. L3-only policy
4. Allow-all policy (по умолчанию)
5. DROP policy

Источник: bpf/lib/policy.h

	QPS	P99.9 (ms)	P99.99 (ms)	P100 (ms)
No policy (default)	6413.8	10.9	12.8	14.2
With policy (CCNP)	6665	10	12.4	13

"Trip.com: First Step towards Cloud Native Security"

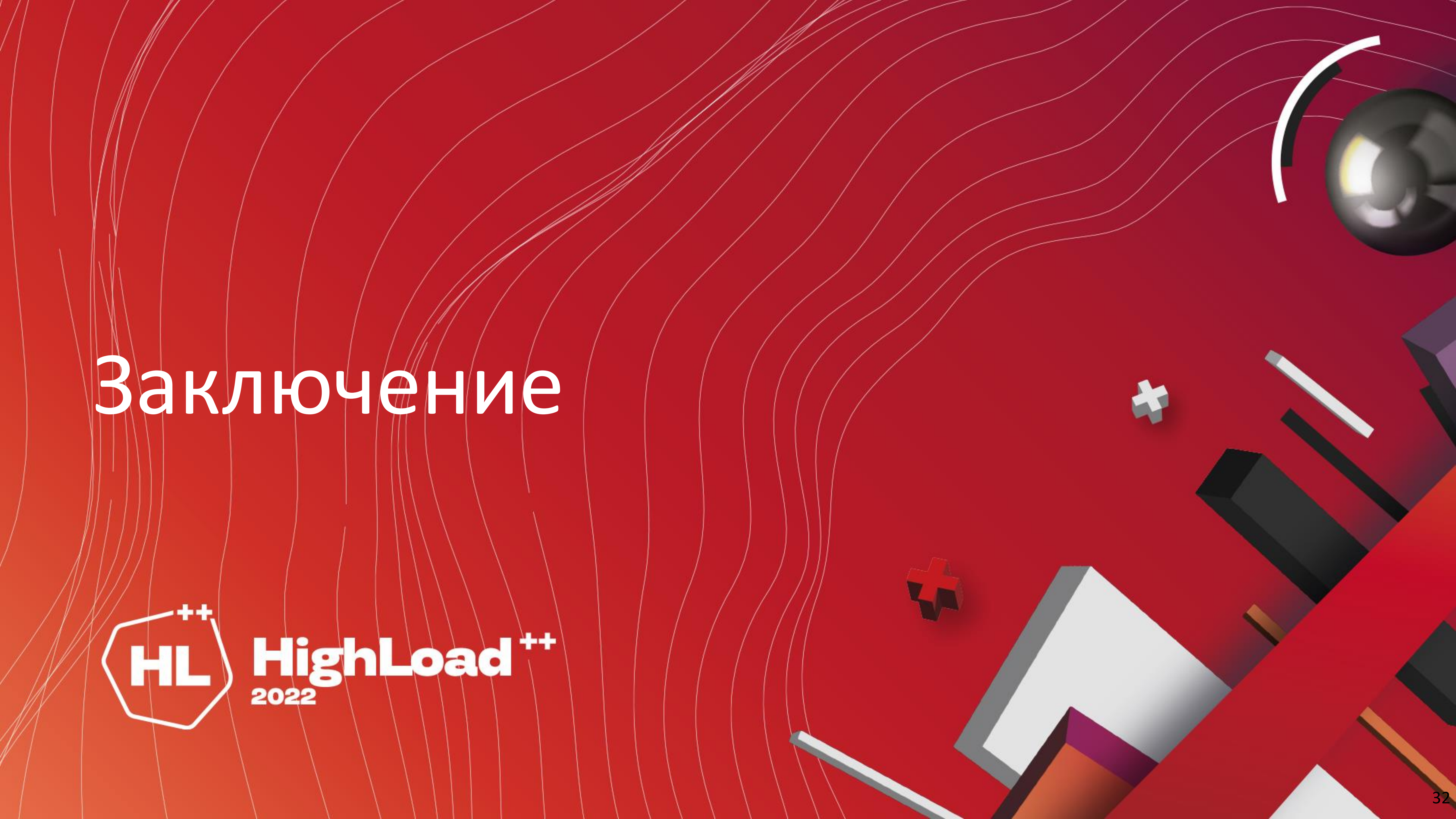


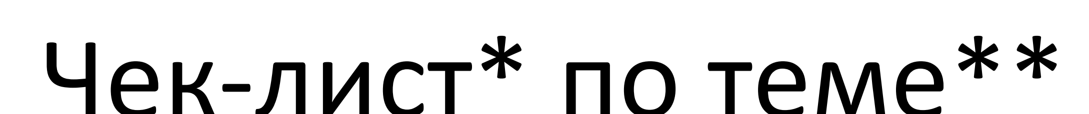


Преимущества NetworkPolicy

- Ускоряем (не замедляем) работу сети
- Организуем микросегментацию сети
  - ZeroTrust
- Инвентаризируем сетевое взаимодействие через декларативное описание
- Прозрачные ограничения для Dev-, Ops-, Sec-команд




Заключение
HL
HighLoad++
2022



Чек-лист* по теме**



К чему надо стремиться при использовании Kubernetes-кластеров:

- Multitenancy
- Policy Engine
- Специализированные ОС хоста для контейнеров
- Tiny/slim/minimal базовые образы
- Network Policy

* Не смотрите на это как на серебряные пули

** Безопасность Kubernetes шире: AuthN/AuthZ, Runtime Security, Audit Logging, Compliance, … !





Разными словами об одном



- Orchestration
  - SOAR
- Declarative approach
  - Policy\Security-as-Code
- Unification
  - Templating, Immutable infrastructure
- Optimization
  - Attack surface reduction
- Micro segmentation
  - Zero Trust



Dai Zovi's Law:

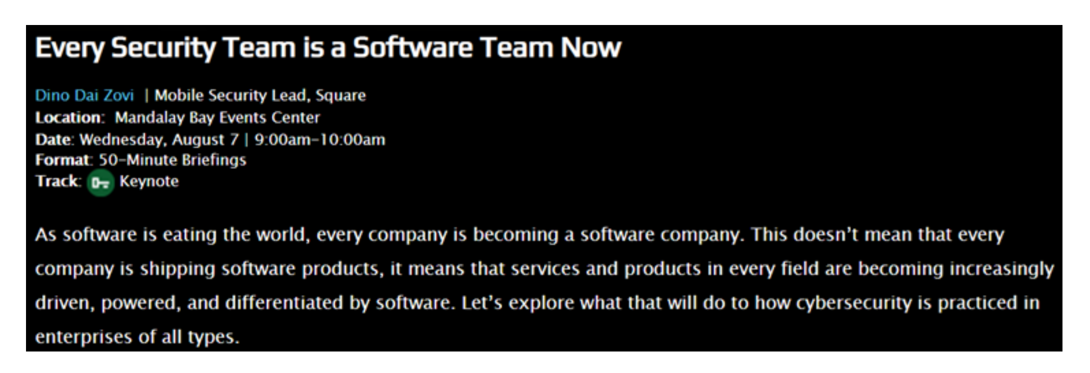
The quality of your organization's security will mirror the quality of the communication between its engineering and security functions.



Dino A. Dai Zovi @dinodaizovi · 4 ч.
100% this. It can get nuanced in specific definitions of particular words, but I think of Security as one dimension of Quality alongside Availability, Performance, etc. Helping increase engineering maturity in ways that benefit Quality also help you improve Security (e.g. CI/CD).



Security is a subset of Reliability: if your designs have no reliability concepts then you will have security issues.
...



Every Security Team is a Software Team Now

Dino Dai Zovi | Mobile Security Lead, Square
Location: Mandalay Bay Events Center
Date: Wednesday, August 7 | 9:00am-10:00am
Format: 50-Minute Briefings
Track: Keynote

As software is eating the world, every company is becoming a software company. This doesn't mean that every company is shipping software products, it means that services and products in every field are becoming increasingly driven, powered, and differentiated by software. Let's explore what that will do to how cybersecurity is practiced in enterprises of all types.





Парадигма DIE

DIE:

- Distributed
- Immutable
- Ephemeral

#RSAC

The DIE Triad



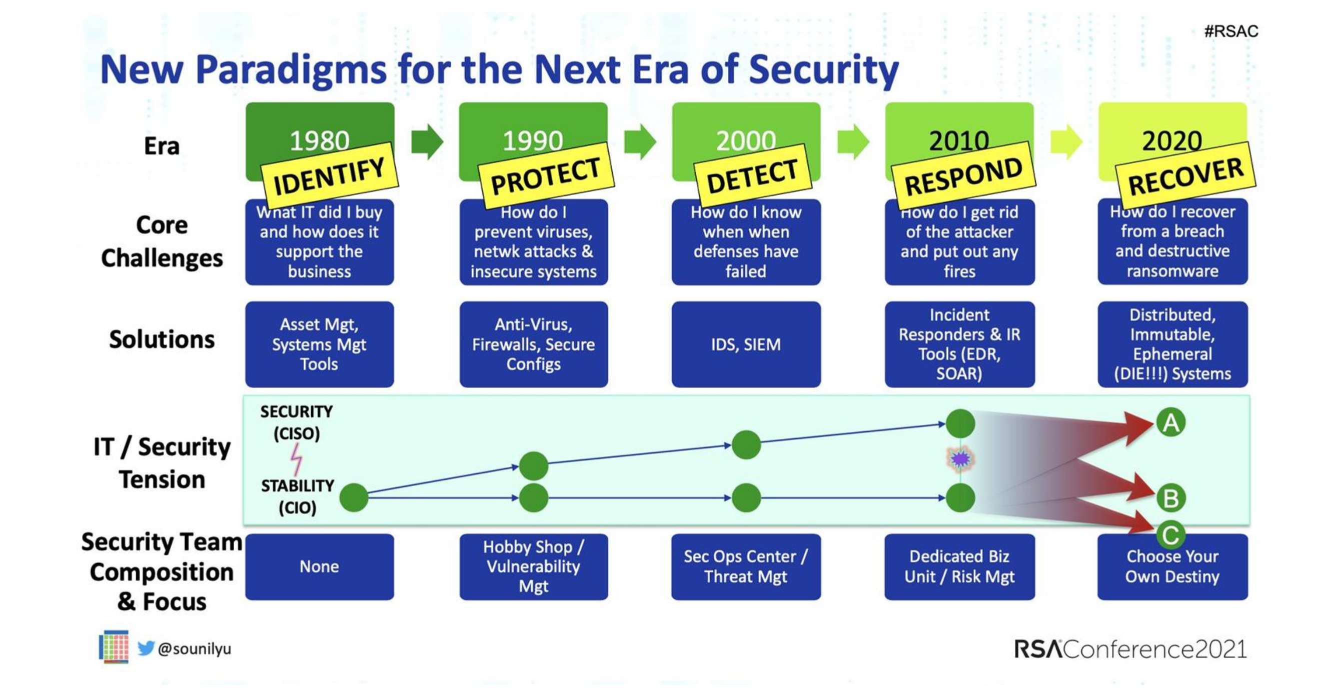








RSAConference2021



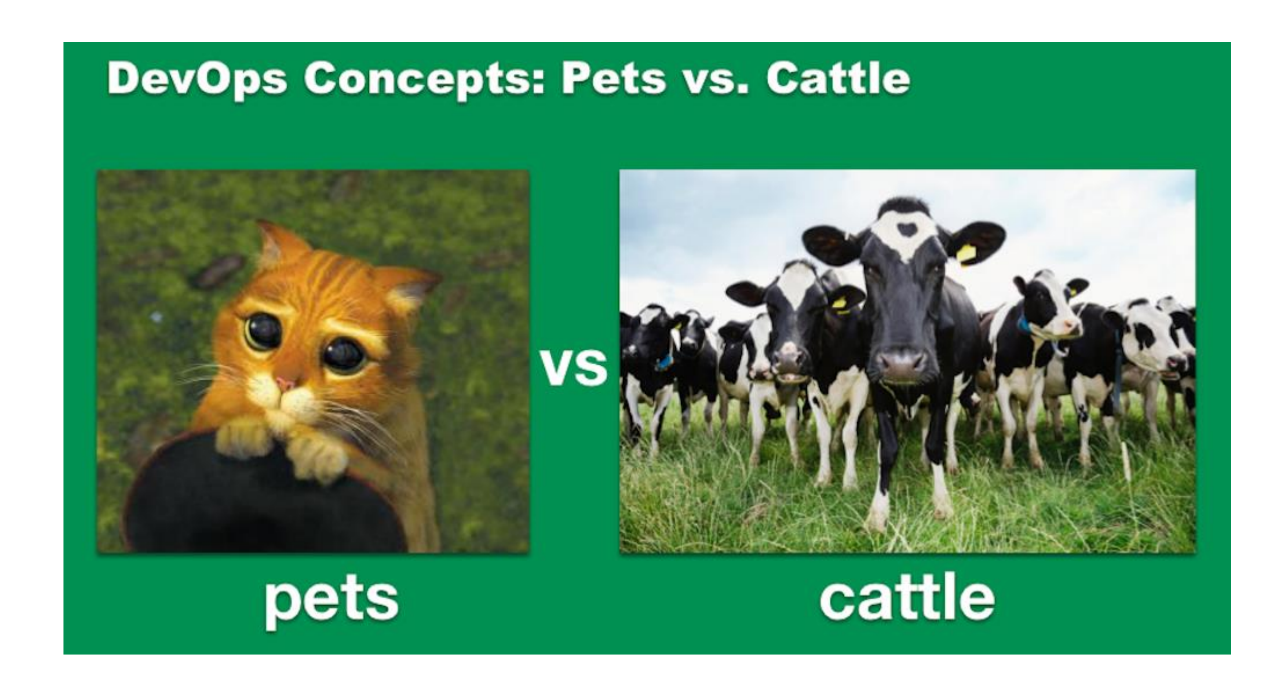
Era	1980 IDENTIFY	1990 PROTECT	2000 DETECT	2010 RESPOND	2020 RECOVER
Core Challenges	What IT did I buy and how does it support the business	How do I prevent viruses, netwk attacks & insecure systems	How do I know when defenses have failed	How do I get rid of the attacker and put out any fires	How do I recover from a breach and destructive ransomware
Solutions	Asset Mgt, Systems Mgt Tools	Anti-Virus, Firewalls, Secure Configs	IDS, SIEM	Incident Responders & IR Tools (EDR, SOAR)	Distributed, Immutable, Ephemeral (DIE!!!) Systems
IT / Security Tension					
Security Team Composition & Focus	None	Hobby Shop / Vulnerability Mgt	Sec Ops Center / Threat Mgt	Dedicated Biz Unit / Risk Mgt	Choose Your Own Destiny

"[New Paradigms for the Next Era of Security]"





Cattle and Pets



Our best countermeasure is to **avoid pet creation** (that requires CIA) and **promote cattle creation** (built to DIE)





Заключение

- Минимализм снова в моде
- Унифицируйте свои окружения
- Боремся со сложностью систем
- ИБ идет к Self-Protection System
- ИТ идет к Antifragile System
- ИТ и ИБ идут рука об руку



Спасибо за внимание!

Email: de@luntry.ru
Twitter: @evdokimovds
Telegram: @Qu3b3c
Channel: @k8security



